# Mixed-Signal Design Flow and Example of SAR ADC Design

Mixed Signal Design Flow

System on a chip with distribution

of analog and digital units

Traditional mixed circuit system (e.g. interface for a MEMS sensor)



# Possible design flows

- Analog centric (or analog on top): the analog and digital units are designed with their proper tools and integration is performed using the analog tool.
  - **Digital centric (or digital on top):** the analog and digital units are designed with their proper tools and integration is performed using the (highly automated) digital tool.

# Analog Design Flow and examples of CAD tools



# Digital Design Flow and CADENCE tools



# Mixed Signal Design Flow: Analog Centric Approach



# An Example: Design of a SAR ADC

- High-level behavioural description of the SAR algorithm (e.g. MATLAB)
- Transistor-level design and simulations of the comparator and the CDAC
- HDL description and digital simulations of the SAR control logic
- Mixed-signal simulations of the whole ADC
- Layout of the analog blocks / synthesis and place-and-route of the SAR logic



P. Bruschi – Design of Mixed Signal Circuits

## Analog Design and Simulations (Spectre simulator)



## VHDL description of the SAR digital control block

	Open 👻 🖭	SAR_ctrl_block.vhd ~/wa_umc/vhdl	Save E _ • ×		
	<pre>library IEEE; use IEEE.numeric_std.all; use IEEE.STD_LOGIC_UNSIGNED.all; use IFFF.std logic 1164.all;</pre>				
	entity SAR_ctrl_block is				
	clk_in: in std_logic; cmp_out: in std_logic;				
	<pre>reset_n: in std_logic; sample: out std_logic; clk_cmp: out std_logic;</pre>				
	SAR_ctrl: out std logic_ output_word: out std_log	vector (5 downto θ); jtc_vector (5 downto θ)			
	end SAR_ctrl_block;				
VHDI Code	type T_STATE is (RESET, SAMPLIN)	j, SAR_PHASE_5, SAR_PHASE_4, SAR_PHASE_3, SAR_PHASE_2, SAR_PHASE_1, SAR_F	HASE_0, FINISH);		
	<pre>signal state: I_SIAIE := RESEI; signal aux_word: std_logic_vecto signal SAR_ctrl_1:std_logic_vect</pre>	r (5 downto θ) := (others => 'θ'); cor (5 downto θ) := (others => '1');			
	<pre>signal SAR_ctrl_2:std_logic_vect signal aux_clk:std_logic := '0'; signal clk aux:std logic := '0'</pre>	or (5 downto 0) := (others => '1');			
	begin SAB_ctrl <= SAB_ctrl 1 ;	and SAR etcl 2.			
	<pre>clk_aux &lt;= clk_in and au clk_cmp &lt;= inertial clk_</pre>	aŭx after lns;			
	process (clk_in, reset_r begin	.)			
	1T (rese	t_n='0') then sample <= '0'; SAR_ctrl_1 <= (others ⇒> '0');			
	elsif r:	state <= RESET; aux_clk <= '0'; sing edge(clk in) then			
		<pre>case state is when RESET =&gt; sample &lt;= '0'; SAR ctrl 1 &lt;= (others =&gt; '0');</pre>			
		state <= SAMPLING; aux_clk <= '0'; when SAMDING == samle <= '1';		Digital simulati	ons (XCELIUM)
		SAR_ctrl_1 <= (others => '0');			
evetorm I - Simvision Edit View Explore Format Simulation Wir	ndows Hein				
					1 243 B Send To: To 200 E 200 E
		- ax ax			
rch Names: Signal - 📕 🖍 📶 🗍	Search Times: Value▼				
FimeA▼ = 8,164,787,8(▼ ps▼ P#*				<b>▶ •  1 1 1 1 1 1 1 1 1 </b>	1,244ps + 0 Time: 🖀 8,110,535,099ps : 🗹 🛼 👗 着 🕇
Baseline▼=0					
Cursor-Baseline ▼ = 8,164,787,883ps	TimeA = 8,164,787,883ps		le eeu ann anna		la 200 000 000
Name	8,200,000,000ps	8,400,000,000ps 1000000 1000000 100000 1000000 1000	8,600,000,000ps	8,800,000,000ps 9,000,000ps	
······································					
i reset_n 1					
1					

🚟 Waveform 1 - SimVision

View Explore tin tin 🖌 Search Names: Signal -

<u>F</u>ile <u>E</u>dit

No. TimeA

### Mixed-Signal Simulations (ams simulator)

#### Config view of the testbench

pCell		? <b>ð</b> ×	Global Bindir	igs		78	
		500					
rary. Esercitazione_SAR			Library List:	ist: Faraday			
tb_SAR			View List:	avioral functional systemVerilog schematic veriloga vhdl vhdlams wreal			
v: schematic					0 0		
			S top List:	spe ctre fsm			
pen Edit		Constraint List	st:				
le View Tree View							
l Bindings	Call	View Found		View To Lice	Inherited View List		
ercitazione SAR	CDAC 6bit	schematic		view to use	spectre spice pspice verilog verilogams	Ê	
- ercitazione_SAR	 CDAC_w_buffer_6bit	schematic_v2	schematic_v2		spectre spice pspice verilog verilogams		
ercitazione_SAR	SAR	schematic			spectre spice pspice verilog verilogams		
ercitazione_SAR	SAR_ctrl_block	functional			spectre spice pspice verilog verilogams		
ercitazione_SAR	delay_line	schematic			spectre spice pspice verilog verilogams	=	
ercitazione_SAR	dig_buffer_w_enable	schematic	1		spectre spice pspice verilog verilogams		
ercitazione_SAR	inverter	schematic			spectre spice pspice verilog verilogams		
ercitazione_SAR	latch_SR	schematic			spectre spice pspice verilog verilogams		
ercitazione_SAR	nor	schematic			spectre spice pspice verilog verilogams		
ercitazione_SAR	pass_gate	schematic			spectre spice pspice verilog verilogams		
ercitazione_SAR	strongARM	schematic			spectre spice pspice verilog verilogams		
ercitazione_SAR	tb_SAR	schematic			spectre spice pspice verilog verilogams		
raday	AN2	functional			spectre spice pspice verilog verilogams		
raday	A01125	functional			spectre spice pspice verilog verilogams		
raday	A0125	functional			spectre spice pspice verilog verilogams		
raday	A02225	functional			spectre spice pspice verilog verilogams		
raday	A0I225	functional			spectre spice pspice verilog verilogams		
radav	D BFR BN	functional			spectre spice pspice verilog verilogams		

#### Connect rules

	File	<u>E</u> dit	View	<u>H</u> elp				cādeı	nce
- [	`de1	fine	CONNRU	JLES_18	V_FULL_FAST				
	`de1	fine	CONNRU	JLES_18	V_FULL				
	`de1	fine	CONNRU	LES_18	V_MID				
	`de1	fine	CONNRU	JLES_18	V_BASIC				
	`de1	fine	CONNRU	JLES_FU	LL_FAST				
	`de1	fine	CONNRU	JLES_FU	LL				
	`de1	fine	CONNRU	JLES_MI	D				
	`de1	fine	CONNRU	JLES_BA	SIC				
	`de1	fine	Vsup	1.8					
	`de1	fine	Vthi	1.2					
	`de1	fine	Vtlo	0.6					
	`de1	fine	Vtrhi	`Vthi/	`Vsup				
	`de1	fine	Vtrlo	`Vtlo/	`Vsup				
	`de1	fine	Vlow	0					
	`de1	fine	Tr	0.2n					
	`de1	fine	Rlo	200					
	`de1	fine	Rhi	200					
	`de1	fine	Rx	40					
	`de1	fine	Rz	10M					
	`de1	fine	Vdelta	1	`Vsup/64				
	`de1	fine	Vdelta	a_tol	`Vdelta/4				
	`de1	fine	Tr_de]	lta	`Tr/20				
	`de1	fine	rsuppl	Ly	4				
	`de1	fine	rpull		1.5e3				
	`de1	fine	rlarge	•	9.0e3				
	`de1	fine	rweak		5.5e4				
	det	fine	rmediu	m	3.2e5				
	de	fine	rsmall	L	1.9e6				

## **Mixed-Signal Simulations**



## **Digital synthesis (GENUS)**



## Digital synthesis (GENUS)

#### RTL (Register Transfer Level) Description (GENUS input)



#### Gate Level Description (GENUS output)

module SAR ctrl block(clk in, cmp out, reset n, sample, clk cmp, SAR ctrl, output word); input clk in, cmp out, reset n; output sample, clk cmp; output [5:0] SAR ctrl, output word; wire clk in, cmp out, reset n; wire sample, clk cmp; wire [5:0] SAR ctrl, output word; wire [5:0] SAR ctrl 2; wire [5:0] aux word; wire [3:0] state; wire [5:0] SAR ctrl 1; wire UNCONNECTED, UNCONNECTED0, UNCONNECTED1, UNCONNECTED2. UNCONNECTED3, UNCONNECTED4, UNCONNECTED5, UNCONNECTED6; wire UNCONNECTED7, UNCONNECTED8, UNCONNECTED9, UNCONNECTED10, UNCONNECTED11, UNCONNECTED12, UNCONNECTED13, UNCONNECTED14; wire UNCONNECTED15, UNCONNECTED16, n 0, n\_1, n\_2, n\_3, n\_5, n\_6; wire n 7, n 8, n 9, n 10, n 11, n 12, n 13, n 14; wire n 15, n 16, n 18, n 19, n 20, n 21, n 23, n 24; wire n\_27, n\_29, n\_31, n\_32, n\_33, n\_34, n\_35, n\_36; wire n 37, n 38, n 39, n 40, n 41, n 42, n 43, n 44; wire n 45, n 46, n 47, n 48, n 50, n 51, n 52, n 53; wire n 55, n 56, n 57, n 58, n 59, n 60, n 61, n 62; wire n 63, n 64, n 68; DBFRBN \SAR ctrl 2 reg[2] (.RB (reset n), .CKB (clk in), .D (n 64), .Q (SAR ctrl 2[2]), .QB (UNCONNECTED)); DBFRBN \SAR\_ctrl\_2\_reg[4] (.RB (reset\_n), .CKB (clk\_in), .D (n\_62), .0 (SAR ctrl 2[4]), .0B (UNCONNECTED0)): DBFRBN \SAR ctrl 2 reg[3] (.RB (reset n), .CKB (clk in), .D (n 60), .Q (SAR\_ctrl\_2[3]), .QB (UNCONNECTED1)); DBFRBN \aux\_word\_reg[4] (.RB (reset\_n), .CKB (clk\_in), .D (n\_58), .Q (aux word[4]), .QB (UNCONNECTED2)); DBFRBN \SAR\_ctrl\_2\_reg[1] (.RB (reset\_n), .CKB (clk in), .D (n 63), .Q (SAR ctrl 2[1]), .QB (UNCONNECTED3)); ND3S g1595 2398(.I1 (n 56), .I2 (n 51), .I3 (n 52), .0 (n 64)); DBFRBN \SAR ctrl 2 reg[5] (.RB (reset n), .CKB (clk in), .D (n 57), .Q (SAR ctrl 2[5]), .QB (UNCONNECTED4)); OR2S g1601 5107(.I1 (n 42), .I2 (n 53), .0 (n 63)); OAI112HS g1590 6260(.A1 (n\_59), .B1 (n\_21), .C1 (n\_50), .C2 (state[1]), .0 (n 62)); DBZRBN \output word reg[1] (.RB (reset n), .CKB (clk in), .D (aux word[1]), .TD (output word[1]), .SEL (n 61), .Q (output word[1]), .OB (UNCONNECTED5));

# Automatic layout generation (Automatic Place and Route, P&R)



# Automatic Place-and-Route (INNOVUS)

Empty spaces between cells are covered by "filler" elements in the final layout. Fillers includes n-wells and other layers that improve continuity between cells



Layout imported in Virtuoso

